

《C/C++ 学习 指南》

第27.1讲：重载算术操作符

作者：邵发 QQ群：417024631

官网：<http://www.afanihao.cn/>

习题：<http://www.afanihao.cn/kbase/>

本书配套纸质教材，具体信息见官网。

C/C++学习指南 邵发 www.afanihao.cn

引 例

用一个类Fraction，用于表示 $\frac{2}{3}$ ， $\frac{87}{890}$ 这种分数。

```
class Fraction
{
public:
    Fraction(int n ,int d): num(n), den(d)
    {
    }
    int num; // 分子
    int den; // 分母
};
```

引例

定义两个数

```
Fraction fa(2, 3); // 2/3
Fraction fb(3, 5); // 3/5
```

按理来说，两个数是可以进行算术运算的，例如

```
int a = 2;
int b = 3;
int c = a + b;
```

那么，对于我们的“分数”这种类型，最好也能够支持加减乘除的算术运算。

目标效果：

```
Fraction fc = fa + fb;
```

（显然，默认情况下自己的class是不支持算术运算的）

重载算术运算符

使用**重载运算符**的语法，可以使得自己定义类支持加减乘除等算术运算符。

第一种方式：类操作符

```
// 重载加号操作符
Fraction operator + (const Fraction& other)
{
    Fraction result;
    // 本对象和other运算，得出结果并返回
    return result;
}
```

重载算术运算符

其中，

`operator +` 是要重载的操作符，形式上类似于成员函数

注意：

- (1) 重载操作符的形式上像一个函数，但它不是函数
- (2) 名称：`operator +`，这个是固定不变的
- (3) 返回值：类型不变，总是为该对象的类型
- (4) 参数：基本上也不变
- (5) 受`public/protected/private`的限制

在 `fc = fa + fb` 语句中，`operator+`被调用

重载算术运算符

另一种写法：重载全局操作符

默认地，操作符 `+` 已经支持了 `double + double`, `int + int`等等，现在要求重载一下，支持 `Fraction + Fraction`

即，已经存在

```
int operator + (int a, int b);
```

```
double operator + (double a, double b);
```

现在要写一种

```
Fraction operator+ (const Fraction&a , const Fraction& b);
```

重载算术运算符

// 重载全局操作符

```
Fraction operator + (const Fraction& a, const Fraction& b)
{
    Fraction result;
    result.den = a.den * b.den;
    result.num = a.num * b.den + a.den * b.num;
    return result;
}
```

通常，要将全局操作符的重载声明为类的朋友

(当num/den为private的时候，必须将此重载的操作符声明为friend)

注意

(1) 当类操作符和全局操作符同时被重载时，类操作符被优先调用。

(就近原则)

(2) 各种重载操作符，其形式基本是固定的。(名称、参数、返回值，基本上都是固定写法)

小结

1. 介绍了算术运算符重载的方法，给出加号(+)的示例。其他的操作符(-, * , /)留作练习。
2. 重载操作符的形式基本上是固定的。
(记忆为主，理解为辅，一般用不上)